ESMM290 Architectures for Software Systems

Assignment 2: Call-Return and Tiered Architectures

Brown team:

Nikita Moshensky
Svetoslav Petkov
Kamen Jordanov

Table of Contents

Business context

EEP is B2B company that supplies exotic plants. EEP experience extraordinary growth since it's funding.

Orders are made through calling EEP's order center. Software order tracking system is used to manage orders. Call center takes orders by phone on site and enter them into database. Shipping personnel checks pending orders, prepare them and after shipping marks them as shipped. All users at site are trusted users. Software system has several applications:

- for call center personnel
- for shipping personnel
- inventory management program

EEP needs to expand from a single collocated center of operations to a distributed operational model. Orders have to be taken at a variety of sites not located at the main EEP facility. During upgrade period, the system must remain in operation until the distributed sites are ready to go on-line.

Stake holders

- IT manager
- Order personnel
- Shipping personnel
- EEP owners
- Clients
- Developers

Architectural Drivers

High-level Function Requirements

The high-level functional requirements are sorted in the order of the importance to the users goal. In this section, the set of high-level functional requirements are ranked in three levels:

- high – critical requirements to the system which are the must-have to satisfy the client of the project
- medium – requirements that have significant impact to the system, but can be

postponed for the future
- low – requirements with low impact on the system. Those requirements only complement the critical once. Without those requirements all intended functionality can be performed

ID	High-level Function Requirements	Priority
1	The system shall provide a software platform that allow users to submit remotely order entries	high
2	The system shall provide a software platform that sits on top of the old system	high
3	Logging feature for tracking user activities	high
4	The system shall allow logs monitoring by IT manager only	high
5	System shall be extendable	medium

Quality Attributes

The following are the quality attribute scenarios in context of the important system qualities they address. They are ordered in decreasing order of their priorities.

ID	QAS1
Raw quality attribute	Secure remote order entry
Source of stimulus	Order entry personnel
Relevant environmental conditions	During runtime execution
Architectural elements	Order application, database
System response	Form for authentication should be present
Response measure	Application has to connect with database and check credentials on success allow personnel to place order. On fail application should exit.
Importance	High

ID	QAS2
Raw quality attribute	IT manager usability accessing user logs
Source of stimulus	IT manager
Relevant environmental conditions	Logs monitoring
Architectural elements	Logs reading application

System response	Generate user activity logs
Response measure	Generate logs on demand with one-two clicks only
Importance	Low

ID	**QAS3**
Raw quality attribute	On fail security authentication
Source of stimulus	Order entry personnel
Relevant environmental conditions	During runtime execution
Architectural elements	Order application, database
System response	Make log in database
Response measure	Track fail count and if more than 3 authentications made in less than 10 minuets, warn IT management.
Importance	Low

Business and technical constraints

ID	Type (Business/Technical)	Constrain	Impact
TC1	Current system has to be operational while implementing new features	Technical	We need to implement small changes to the system, in order to keep the system up and running.
TC2	Limited resources	Technical	The team may spend time learning new technologies to be able to design and build the system, such as Java, MySQL and so forth.
BC3	A complete system must be delivered before 31 January 2013	Business	Adequate time may not be available for all project required activities. We may have to take short cuts.

Question 1

According to the business context, what are the key architectural drivers of the system and what are their relative priorities?

Please see above descriptions.

Please describe those factors that motivated the changes to the system and describe how those factors influenced your design decisions.

Factors that motivated changes to the system are related to extraordinary growth of EEP business and need for remote order processing. Also old system had to be operational while we was implementing the new one. Taking in to account that old architecture was shared information system we decided to keep it, because it promotes scaling – easy to add accessors.

Question 2

System Architecture

To fulfill the above business requirements, the system is based on shared information client-server style.

We have extended the current data base to store needed data. Secured the remote access to the system through ordering application, which enables user to connect remotely to database. We log to database users activity and provide IT Manager with tool for monitoring it.

On figure 1 you can see systems System Architecture – Dynamic Perspective:

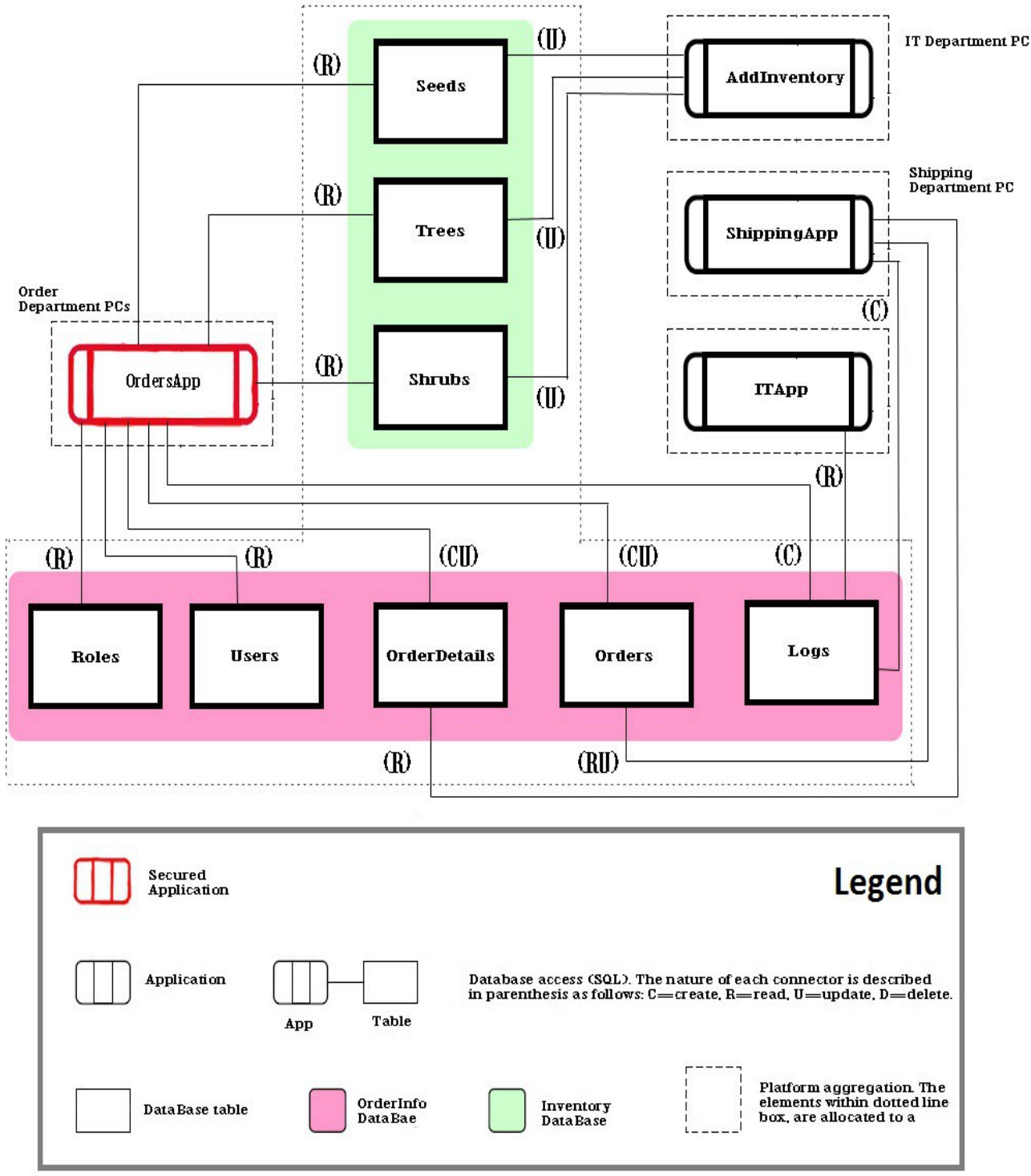


Figure 1: System Architecture – Dynamic Perspective

On figure 2 to is presented data model view:

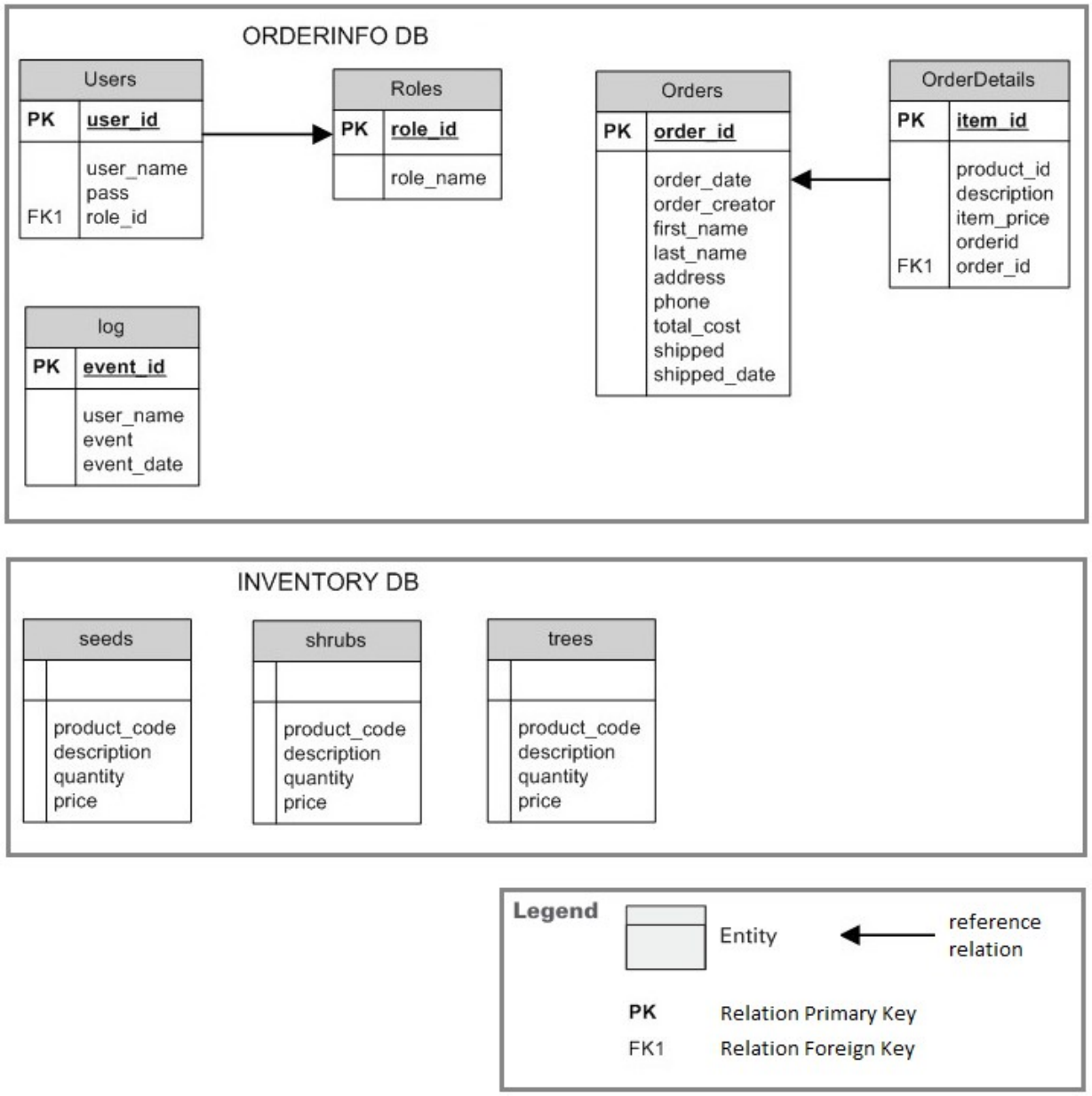


Figure 2: Data model view

On figure 3 you can see ordering case:

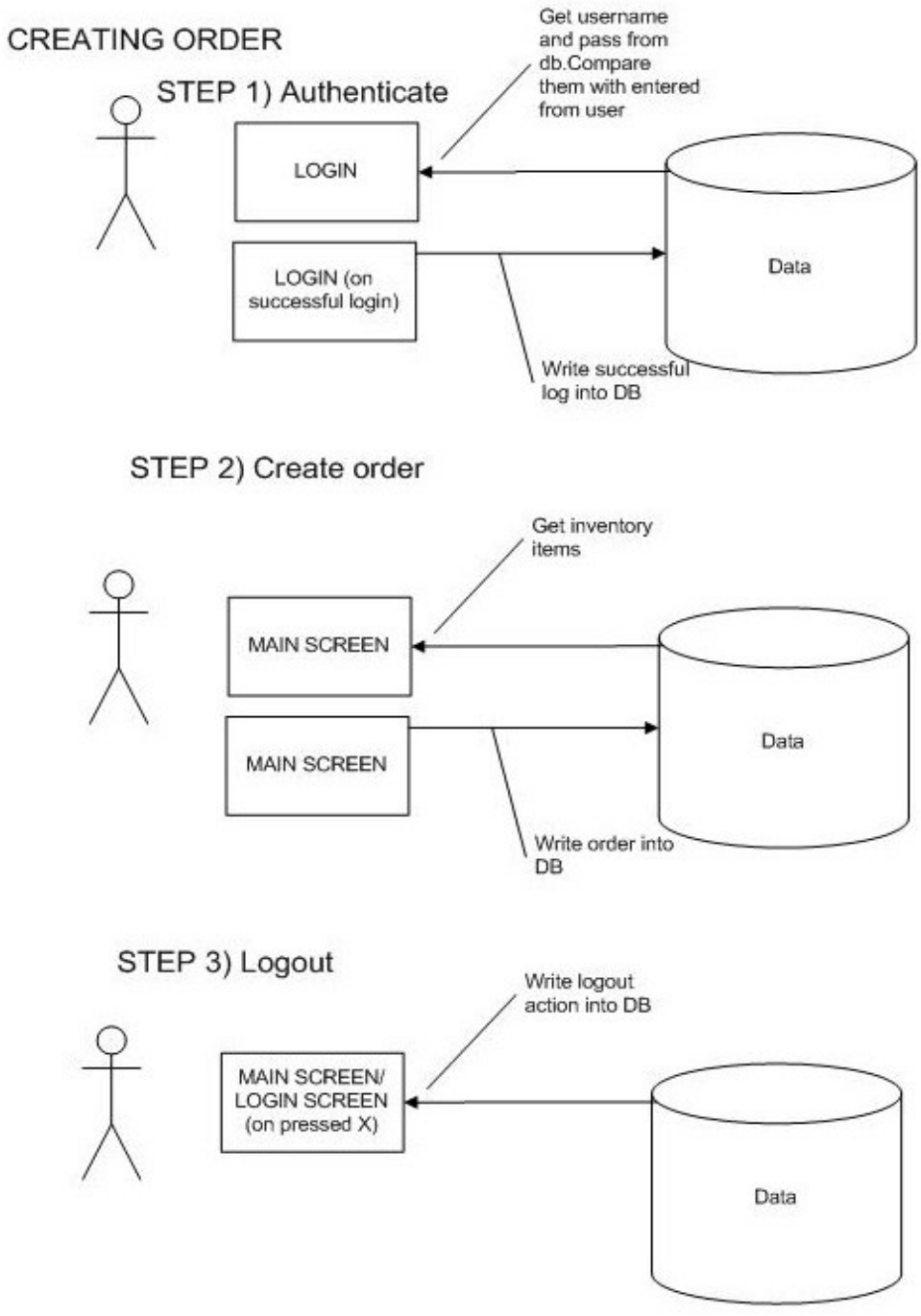


Figure 3: Ordering case

How well current system supports the business goals int terms of important architectural drivers

Order personnel can make remote order entries, which will support rapid business growth. With new features employees will be more efficient and IT manager have good look over each process due to logs.

The system sits on top of the previous software platform and new order entrie application is secured according to quality attributes requirements.

Tradeoffs and other possible solutions

We have tried to keep as little as possible changes to the system. Because of that we have made tradeoffs as follow:

1. Security – directly communicate with data base, which is not a good practice for remote applications.
2. Re-usability – using tree applications instead of one, which makes system difficult to maintain.
3. Desktop applications – which means that after each system update IT personnel has to install latest versions.

There are other possible solutions that we could adopt. We could introduce middleware in to the system. The software layer that lies between the operating system and applications on each side of a distributed computing system in a network. Services that can be regarded as middleware include enterprise application integration, data integration, message oriented middleware (MOM), object request brokers (ORBs), and the enterprise service bus (ESB).

Figure 4 represent middleware, which allows users to request data from the database using forms displayed on a Web browser and it enables the Web server to return dynamic Web pages based on the user's requests and profile.

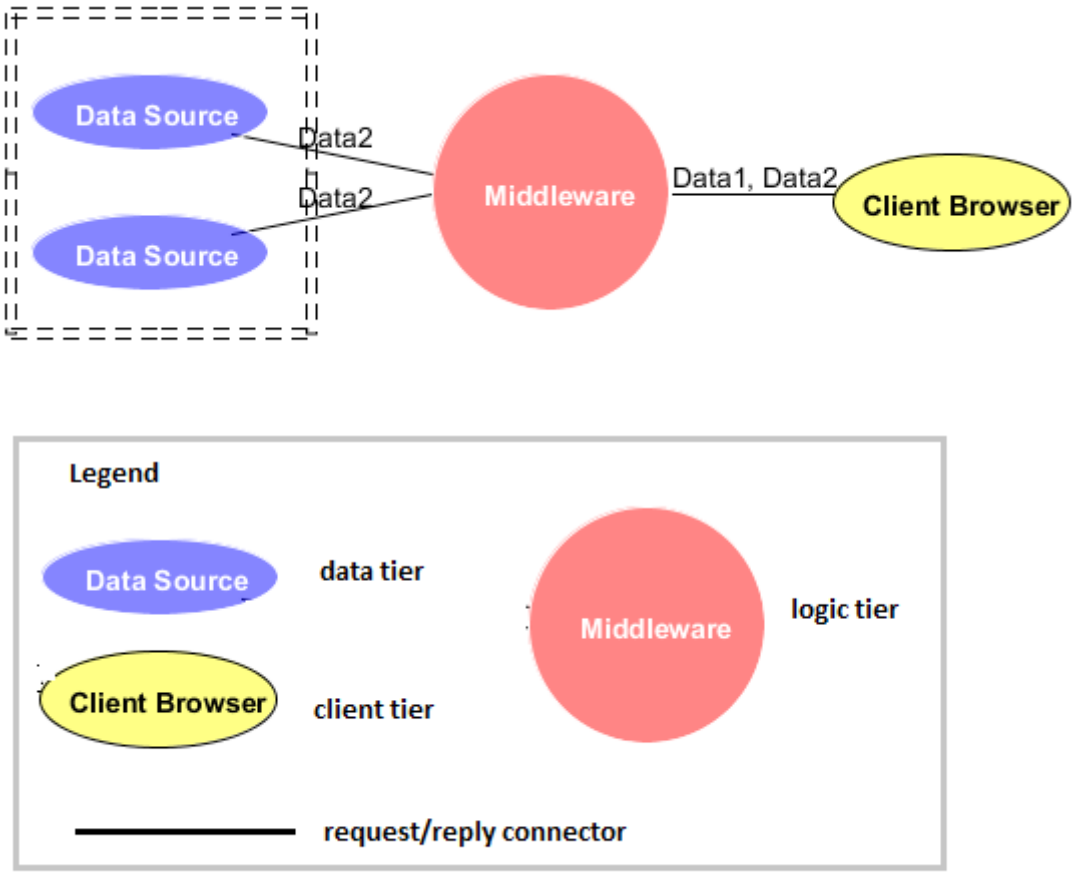


Figure 4: Middleware based on HTTP protocol

Figure 5 represent middleware, above the network and transport layers exemplified by the TCP/IP protocol suite and its sockets interface. Often, middleware products will supply session, presentation and application layer components.

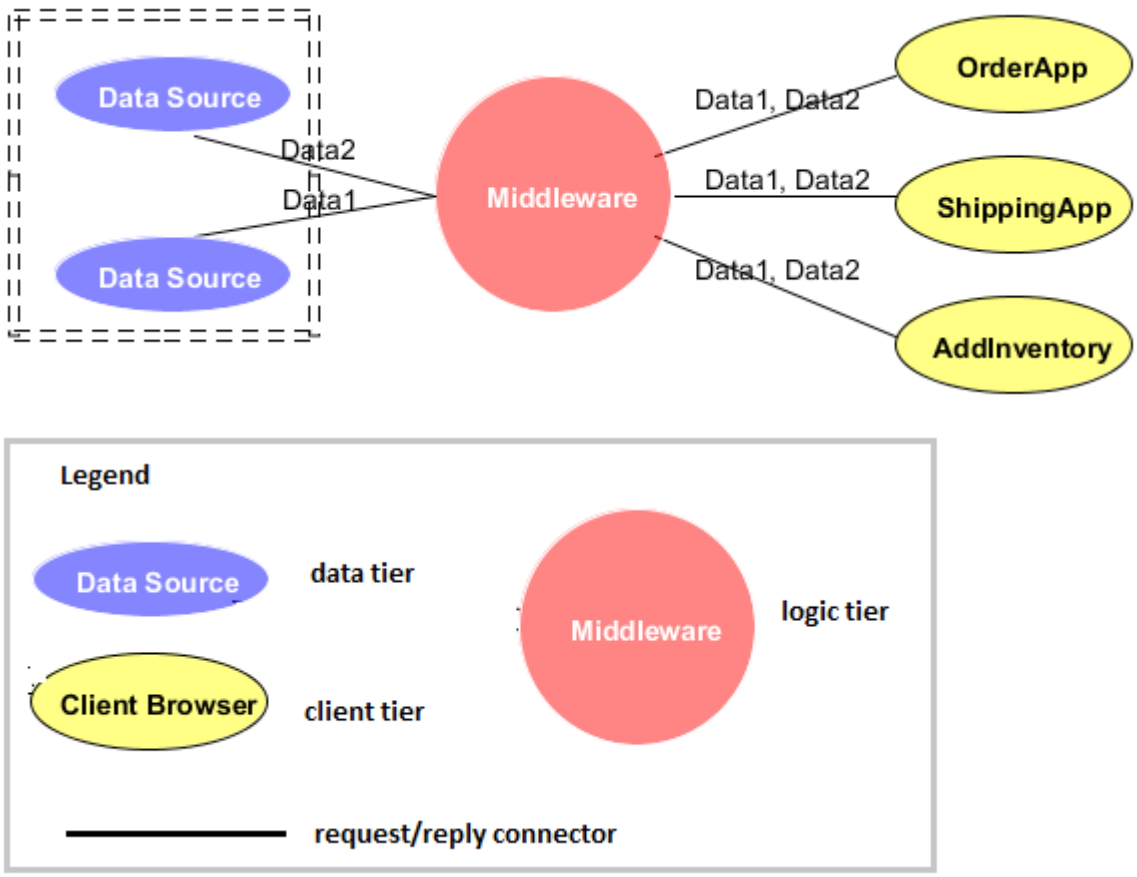


Figure 5: Middleware system based on TCP/IP protocol

Question 3 – Data distribution

How "difficult" would it be in your modified system to distribute the data in the database across different servers given the existing technological constraints and architectural design choices?

How "difficult" would it be in your modified system to replicate the data in the database to address scaling and/or availability concerns given the existing technological and architectural design choices?

Changes above can be done in our modified system. For this to happen we will have to do:

- modify software architecture
- applications may stay intact, but database/server module will have to be modified
- new tier/s and layers will be introduced
- we will require time for developing and testing

- new production and QA servers will have to be bought
- new data base administration resource will be needed

On this figure 6 you can see modified server architecture:

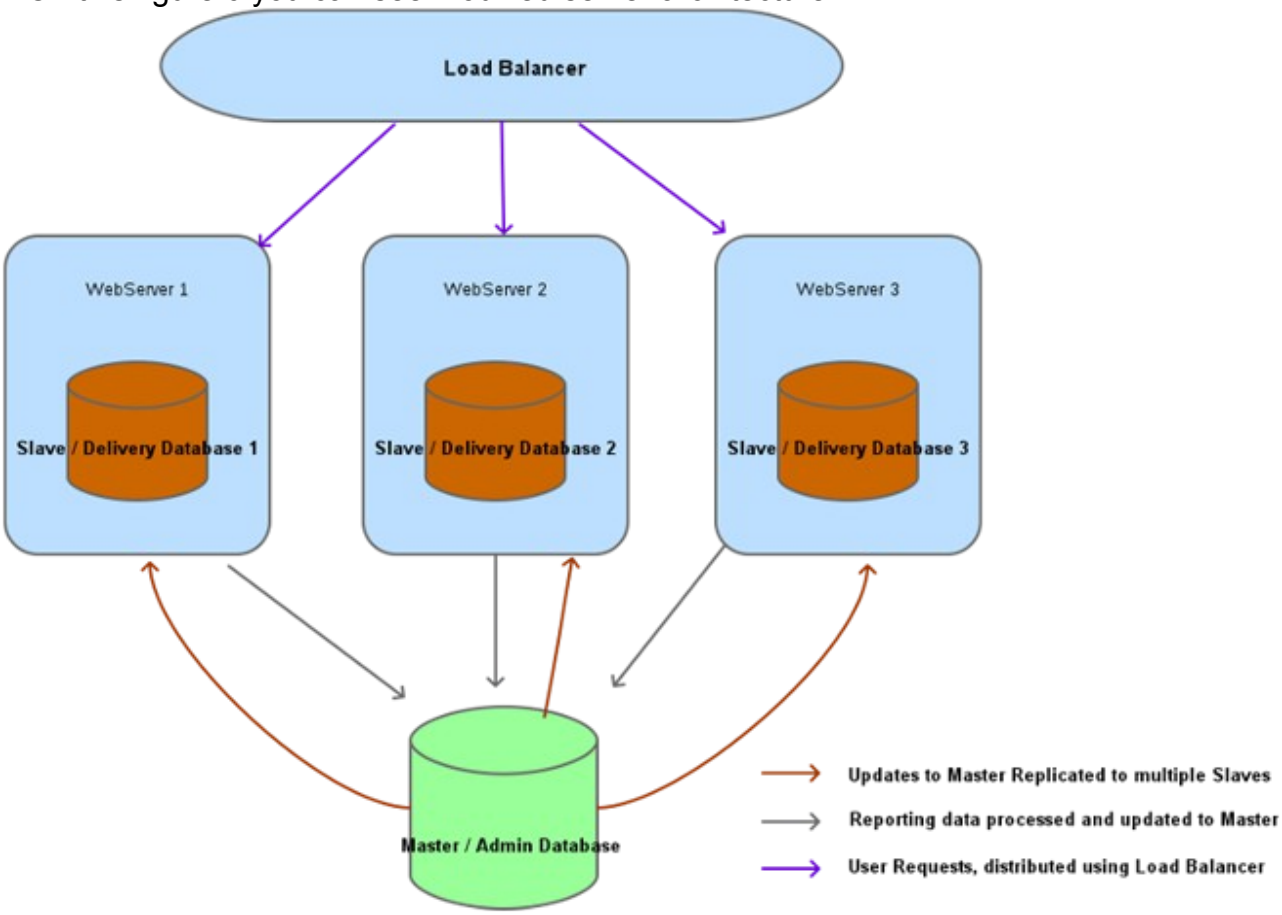


Figure 6: Distribution of database

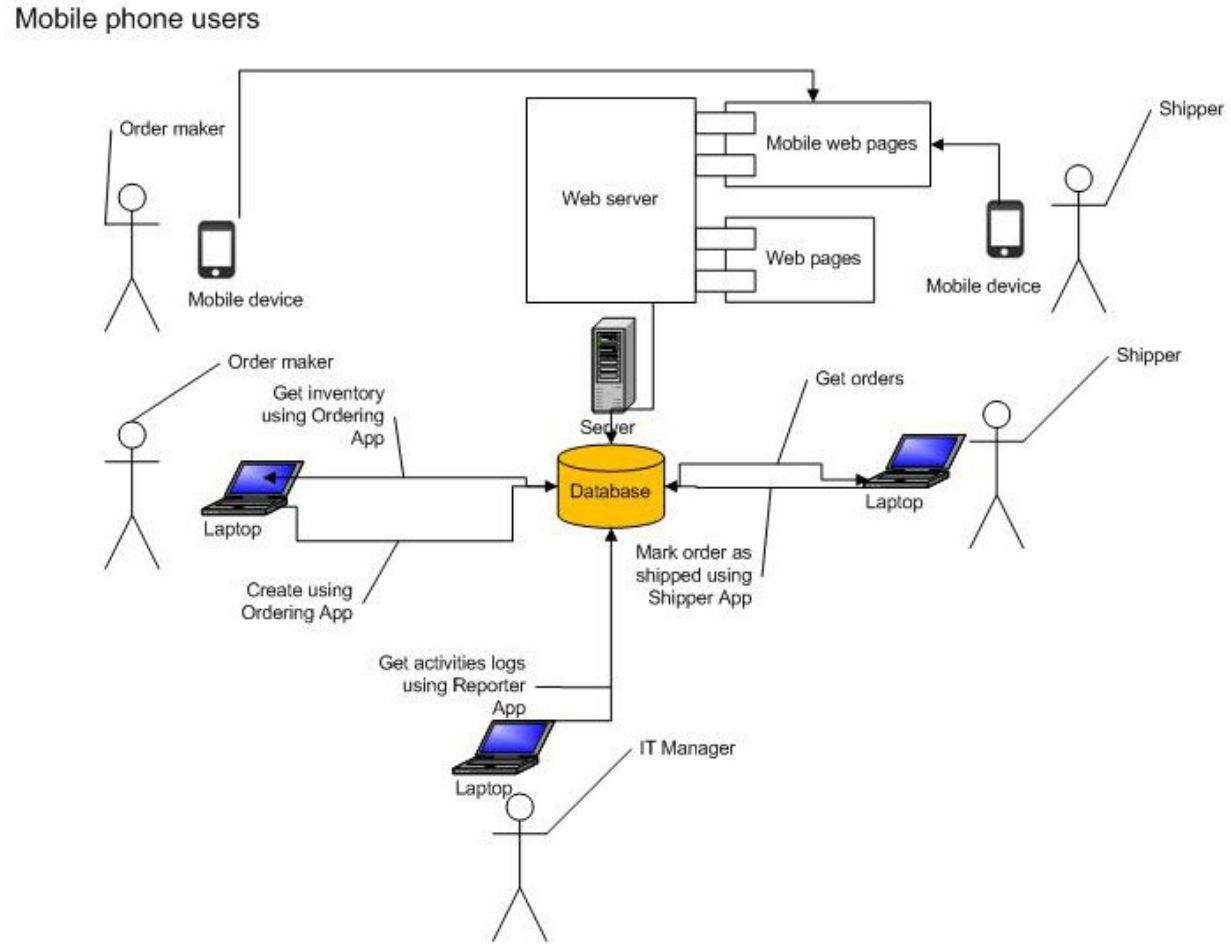
This architecture is based around a series of delivery servers reporting back to a central database. The server farm can be scaled by adding more delivery servers. This implementation provides good redundancy, as a failure of any single server will not disable the system. The load balancer will handle a delivery server outage by distributing the extra load across the other delivery servers. With this architecture we will be able to scale to serve as many requests as we need to.

Question 4 – Submit orders using web browser

How would you modify your upgraded system so that customers could directly submit orders using a web browser? What are the architectural implications of your solution, and how well will your modified system support

this requirement? (Note that you do not have to implement this modification, but you must describe a realistic solution.)

New software architecture:



We will add additional tier – middleware, but only between mobile devices and database. This way old system could continue to function just right. New middleware will implement all needed business logic and serve web pages to mobile devices.

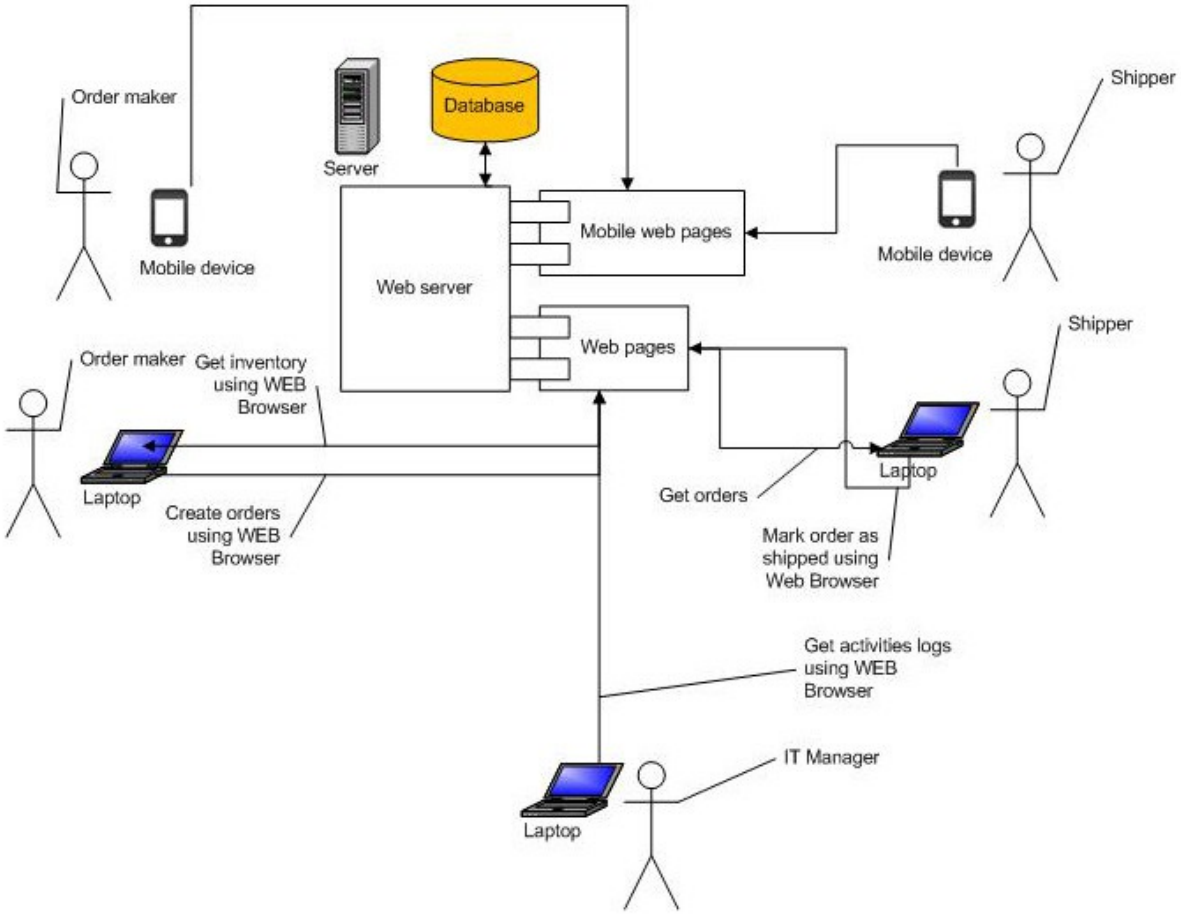
Question 5 – Accommodate the use of mobile devices for the shipping department

How would you modify your upgraded system to accommodate the use of mobile devices for the shipping department personnel? Assume that they will be using mobile devices to indicate when they packed and shipped the orders. In addition to recording the order number and the time and date that the item

shipped, they would also like to record who packed the order. (Note that you do not have to implement this modification, but you must describe a realistic solution.)

New software architecture:

LOT OF USERS –
MIGRATE TO WEB –
NEXT NEXT STEP



Having integrated middleware in previous step, we would continue to replace all old applications with web browser adding needed logic to middleware.